AUDIO TAGGING USING A LINEAR NOISE MODELLING LAYER

Shubhr Singh, Arjun Pankajakshan and Emmanouil Benetos
{s.singh@se17. , a.pankajakshan@ , emmanouil.benetos@}qmul.ac.uk

School of Electronic Engineering and Computer Science, Queen Mary University of London, UK

ABSTRACT

Label noise refers to the presence of inaccurate target labels in a dataset. It is an impediment to the performance of a deep neural network (DNN) as the network tends to overfit to the label noise, hence it becomes imperative to devise a generic methodology to counter the effects of label noise. FSDnoisy18k is an audio dataset collected with the aim of encouraging research on label noise for sound event classification. The dataset contains ∼42.5 hours of audio recordings divided across 20 classes, with a small amount of manually verified labels and a large amount of noisy data. Using this dataset, our work intends to explore the potential of modelling the label noise distribution by adding a linear layer on top of a baseline network. The accuracy of the approach is compared to an alternative approach of adopting a noise robust loss function. Results show that modelling the noise distribution improves the accuracy of the baseline network in a similar capacity to the soft bootstrapping loss.

Index Terms— Audio tagging, noisy labels, noise adaptation layer, noise robust loss function.

1. INTRODUCTION

Audio tagging refers to the classification task which involves predicting the presence of one or more acoustic events in a particular audio recording. Humans are able to perform this task effortlessly, however modelling this cognitive process through computational methods is non-trivial [1] and is an active research area which has received increased attention in recent years. The majority of current approaches to the problem involve supervised training of a deep neural network (DNN) on the labels associated with each audio recording [2],[3]. The basic assumption that comes along with these approaches is that the label provided with the audio recording is correct, i.e. the presence of the acoustic event associated with the label corresponding to the audio recording has been manually verified. This assumption does not always hold true as manual verification of data labels is a costly affair, effectively limiting the size of the data sets.

As described in [4], a large amount of audio data accumulation comes at the cost of imprecise labels, especially in cases such where labels have been inferred based on user provided metadata, i.e. tags. As the labels are noisy, there is a high probability of misleading information, which in turn subverts the training of a DNN. Recent studies [5], [6] have shown that the generalization capability of a DNN reduces on datasets with noisy labels, i.e. the model overfits the training data.

In this paper, we explore the FSDnoisy18k dataset [4] for sound event classification, which contains a small subset (10%) of accurately labelled data and a large subset (90%) consisting of data samples with noisy labels. We use a linear noise distribution modelling layer approach and compare its performance on the test set with that of the baseline model and also with the soft bootstrapping loss function approach [7]. Both approaches have been implemented for the problem of noisy labels in computer vision [6], [7], however to the authors' knowledge, experiments on audio data with a linear noise modelling layer is yet to be explored. We adopt the MobilenetV2 architecture as our baseline model without any pretrained weights and train the network on the training set of FSDnoisy18k. The model weights with the least categorical cross entropy loss (CCE) on the validation set are selected and a dense layer with softmax activation is placed on top of the network and re-trained on the training set. The purpose of the re-training is to learn the weight parameters of the noise modelling layer. The noise modelling layer is removed during prediction on the test set. We observed that the noise modelling layer approach improves the accuracy of the baseline network on test set by approximately 2% and the soft bootstrapping loss function approach improves the accuracy score of the baseline network by approximately 1.5%.

The paper is organized as follows. Section 2 introduces related work, Section 3 discusses the characteristics of the dataset used in this paper, Section 4 discusses the type of label noise present in the dataset, Section 5 details the MobilenetV2 architecture, Section 6 discusses the two approaches to the problem, Section 7 covers the experimental setup and the evaluation metrics adopted for the paper, Section 8 discusses the results, and Section 9 concludes the paper and discusses future work.

2. RELATED WORK

Various approaches have been proposed to deal with the problem of noisy labels in the computer vision domain. One line of approach involves modelling the distribution of noisy and true labels using DNNs [6], [8]. The noise model is used to infer the true labels from the noisy labels. These methods explicitly require a small subset of the data with trustworthy labels. The true label is considered to be a latent random variable and the noise processes is modelled by a linear layer with unknown parameters. Reasoning for using a linear layer is explained in section 6. The expectation-maximization (EM) algorithm [9] is applied to find the parameters of both the linear layer and the neural network to find the correct labels.

A different line of approach involves the soft bootstrapping loss function [7], where the target label is dynamically updated to a convex combination of the original noisy label and the label predicted by the model at that point in time. The updated target label is used for calculating the CCE loss against the predicted label. The underlying concept behind the custom loss function is that label noise causes high deviation between the label predicted by the model and the observed label, due to which the loss is artificially inflated and

AP is supported by a QMUL Principal's studentship. EB is supported by RAEng Research Fellowship RF/128 and a Turing Fellowship.

to reduce this loss component, the model memorizes the noisy label, hence to rectify this situation, current model prediction is added as a consistency objective to the observed label and as learning progresses during training, the predictions of the network tend to become more reliable, negating the effect of label noise to an extent. The soft and hard bootstrapping methods have been evaluated in [4] using audio data and have shown to improve the accuracy of the network. To the authors' knowledge, [4] is the only work in the audio domain with a soft bootstrapping loss function implemented for noise robustness.

3. DATASET

The FSDnoisy18k dataset [4] consists of audio recordings unequally distributed across 20 acoustic event classes: Acoustic guitar, Bass guitar, Clapping, Coin-dropping, Crash cymbal, Dishes pots and pans, Engine, Fart, Fire, Fireworks, Footsteps, Glass, Hi-hat, Piano, Rain, Slam, Squeak, Tearing, Walk, Wind, and Writing.

Audio recordings are of varying lengths, ranging from 30ms to 300ms and can be broadly divided into two categories of labels - noisy and clean. The proportion of noisy/clean labels in terms of the number of audio recordings is 90%/10% and in terms of duration, the proportion is 94%/6%.

The training set consists of 17,585 clips whereas the test set comprises 947 clips. The test set has been formed entirely from the clean label dataset, with the remaining data forming the training set. The number of clips per class ranges from 51 to 170 in the clean subset and 250 to 1000 in the noisy subset. The dataset contains a single label per audio recording.

4. LABEL NOISE

Label noise can either be synthetically injected into the dataset [7] or can already be present in the dataset (real world noise). FSDnoisy18k [4] contains real world label noise since the class label have been annotated based on the associated user tags from freesound [10]. Before elucidating the label noise types, it is important to discuss the data collection and annotation process adopted for the dataset. First a number of freesound user-generated tags were mapped to classes based on Audio set ontology definition [11], post which, for each class, audio clips were selected from freesound, tagged with at least one of the selected user tags. This process generated a number of potential annotations, each of which indicated the presence of a particular class in the given audio recording. The potential annotations were verified via a validation task hosted on FSD online platform [10], where users were required to validate the presence or absence of each of the potential annotations by choosing one of the following options [10]:

1. Present & Predominant (PP) - The sound event is clearly present and predominant.
2. Present but not predominant (PNP) - The sound event is present, but the audio recording also contains other types of sound events and/or background noise.
3. Not Present (NP) - The sound event is not present in the audio recording.
4. Unsure (U) - Note sure if the sound event is present or not.

The audio recordings with annotations rated as PP by a majority of users were included in the training set with curated labels and the test set. The remaining audio clips are included in the training set with noisy labels. The label noise types found in the dataset can be characterized into the following categories:

1. Incorrect/out of vocabulary (OOV) - The accurate label describing the sound event does not correspond to any of the Audio set [11] classes.
2. Incomplete/OOV - Some audio clips contain acoustic events in addition to their accurate labels, however only one sound event is mentioned in the label since the other sound events do not belong to any of the Audio set [11] classes.
3. Incorrect/In vocabulary (IV) - This type of noise consists of classes which are closely related to each other, (e.g. "wind" and "rain") and have been interchanged.
4. Incomplete/In vocabulary (IV) - Two sound events are co-occurring on the audio recording, despite only a single label reported.
5. Ambiguous labels - It is not clear whether the label is correct or not.

The distribution of label noise types in random 15% of per class data in the dataset is shown in Table1. The analysis of noisy labelled training dataset revealed that approximately 60% of the labels contain one or multiple types of label noise and 40% of the labels are correct [4]. As can be seen from Table1, OOV noise constitutes a major portion of the label noise across different classes, either in form of incorrect labels or incomplete labels.

5. BASELINE MODEL

MobilenetV2 [12] is selected as the baseline model. It builds upon the MobilenetV1 [13] architecture which uses depth wise separable convolution layers as the building block. MobilenetV1 consists of a single convolutional layer followed by 13 separable convolution layers. An average pooling layer follows the last separable convolution layer. In separable convolution, the kernel step is divided into depthwise and pointwise convolution operations. A depthwise convolution acts on each channel independently, post which a pointwise convolution acts across all the channels. This factoring reduces the weights of each layer, making the model compact without loss of accuracy. The MobilenetV2 architecture has two new features on top of its predecessor:

1. Linear bottleneck between layers,
2. Residual connection between the bottlenecks.

A pointwise convolution operation in a separable convolution layer leads to an increase in the number of channels. A linear bottleneck layer does the exact opposite. It reduces the amount of data flowing through the network. The residual connection between the linear bottlenecks work in the same manner as Residual Nets [14], where the skip connection serves to assist the flow of gradients through the network. MobilenetV1 was used as the baseline network for the DCASE 2019 Challenge Task 2 [15], a multi label audio tagging task with a large amount of noisy labelled training data and a small amount of manually curated training data. The test data was free of label noise. The source of the manually curated training dataset and test data was the Freesound dataset [10]. This inspired us to explore the MobilenetV2 architecture for our experiments.

6. METHODS

We explore the approaches proposed in [6] and [4] for our experiments on audio tagging with noisy labels.

Table 1: The table has been adopted from [4] and depicts the distribution of label noise types in a random 15% of the noisy data of FSDnoisy18k. The most predominant noise type is Incorrect OOV which refers to the noise type where user tags were not mapping to any existing class.

Label noise type	Amount
Incorrect/OOV	38%
Incomplete/OOV	10%
Incorrect/IV	6%
Incomplete/IV	5%
Ambiguous labels	1%

6.1. Noise modelling with linear layer

This approach intends to find the latent clean label from the noisy labels [6]. A linear noise modelling layer is added on top of the softmax layer. The parameters of the noise layer are denoted by:

$$\theta(i,j) = p(z = j|y = i). \tag{1}$$

z is the observed noisy label and y is the latent clean label. i and j belong to the class set $\{1, 2....k\}$. This parameter representation denotes the probability of observing a noisy label $z = j$ given the latent true label $y = i$. The equation assumes that the noise generation is independent of the input vector and only depends upon the latent clean labels. This is a simple version of the noise adaption layer and can be implemented using a dense layer with softmax activation.

For the complex version of the noise adaptation layer, given the input vector x, network parameters w, noisy label z, noise distribution parameter θ defined in equation (1) and number of classes n, the probability of observing noisy label z given the feature vector x can be denoted by the equation:

$$p(z = j|x; w, \theta) = \sum_{k=1}^{n} p(z = j|y = i; \theta)p(y = i|x; w) \tag{2}$$

The block diagram of the noise adaptation layer approach is shown in Fig. 1. h in the figure denotes the non linear function $h = h(x)$ applied on the input x. w_{noise} refers to the weights of the noise modelling layer. For our model, which is identical to the simple model proposed in [6], the weights are initialized from the prediction output of the baseline model on the training set and are learnt along with the weights of the neural network (w) during the training phase. The linear noise modelling layer is not used during the test phase.

The experimental procedure is as follows: First, the baseline network (MobilenetV2) is trained on the training set containing both noisy and clean labels. The weights of the model are learnt from scratch during the training phase. The prediction output of the baseline network on the training set is used to initialize the weights of the noise modelling layer which is basically a dense layer with softmax activation. The noise modelling layer is added on top of the baseline network and retrained on the training set to learn the weights of the noisy channel.

The noise modelling layer is removed during prediction on the test set, the reason being that we want to see how the transformed baseline network performs on the test set as compared to the original baseline architecture.

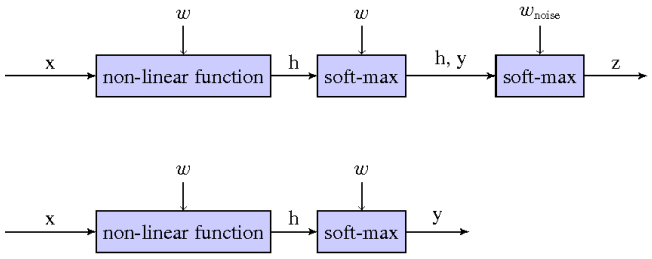

Figure 1: The figure has been adapted from [6]. It is an illustration of the architecture for the training phase (above) and the test phase (below).

6.2. Soft bootstrapping loss

The soft bootstrapping loss was originally introduced in [7] and has been implemented in the audio domain in [4]. The loss function dynamically updates the target labels based on the models' current output. The idea is to pay less attention to the noisy labels, in favour of the model predictions, which are more reliable as the learning progresses. This approach can be expressed by:

$$L_{soft} = -\sum_{k=1}^{n}[\beta y_k + (1-\beta)\hat{y}_k]\log(\hat{y}_k), \quad \beta \in [0,1] \tag{3}$$

$\hat{y}_k$ is the k'th element of the network predictions (the predicted class probabilities), and n is the number of classes. The parameter β is used to assign the weightage of each component in the total loss. The updated target label is a convex combination of the current model's prediction and the (potentially noisy) target label.

7. EXPERIMENTAL SETUP AND METRICS

In this section we discuss the experimental setup and the evaluation metrics adopted for the paper.

7.1. Experimental Setup

Given the nature of the dataset with noisy labels, we are interested in exploring how well the baseline and baseline & linear noise modelling layer would perform on the dataset. The incoming audio is transformed to a 128 band log-mel spectogram using a window size of 1764 (44100(sampling rate) x 0.04(40 ms for a frame)), samples and a hop length of 882 (44100 *0.02(20 ms for overlap)) samples. Since each audio recording is of different length, the duration of each recording is fixed to 2s. The longer recordings were clipped whereas the shorter ones were replicated to obtain a uniform length across the dataset. Both the training set and the test set are scaled using the mean and standard deviation of the training set, post which the class distribution is balanced by oversampling the classes with less samples using the oversampling function from the imblearn library [16].

Data augmentation is also applied as a part of preprocessing. We use mix up data augmentation [17] where new samples are created through a weighted linear interpolation of two existing samples. (x_i, y_i) and (x_j, y_j) are two samples randomly selected from the training set and a convex combination using the parameter $\lambda \in [0, 1]$ which decides the mixing proportions. A new pair of samples (x_k, y_k) is formed using the equations:

$$x_k = \lambda x_i + (1-\lambda)x_j \tag{4}$$

$$y_k = \lambda y_i + (1 - \lambda) y_j \quad (5)$$

The training data is split into training and validation sets with the entire manually verified data used as validation set. The cross entropy loss is used in all experiments except one, where the soft bootstrapping loss is used. An initial learning rate of 0.001 and batch size of 64 samples is used. Each model is trained for 250 Epochs.

The following experimental scenarios are evaluated:

1. Using the baseline network without data augmentation.
2. Using the baseline network with data augmentation.
3. Using the best model from step 2 and adding a dense layer with softmax activation on top of the network. The new network is retrained on the training data with the same setup for training and validation. The weights of the linear layer are initialized using the prediction output from the baseline network. This is in line with the simple model proposed in [6].
4. Training the baseline network from scratch using the soft bootstrapping loss.

7.2. Evaluation Metrics

Classification accuracy was used as the evaluation metric for all the experiments and the results for training, validation and test accuracy have been reported in the results section.

8. RESULTS

Table 2 presents the results of all the experimental approaches mentioned in Section 7.1. Adding a linear noise modelling layer increases the accuracy of the baseline network by approximately 2%. The soft bootstrapping loss function approach also improves the accuracy of the baseline model by approximately 1.5%, indicating that both the approaches are to an extent, helpful in dealing with label noise.

Table 2: Results

Approach	Test Accuracy
Baseline w/o data augmentation	0.649
Baseline with data augmentation	0.667
Baseline with linear noise layer	**0.686**
Baseline with soft bootstrapping loss	0.6825

As can be seen from Fig. 2, the noise modelling approach improves the performance of the baseline network in certain classes such as Coin_dropping, Dishes_and_pots_and_pans,Crash cymbal,Wind,fireworks,Hi-hat and Writing, however the accuracy either decreased or was equivalent to the baseline accuracy for Walk_or_footsteps,Rain,Engine,Glass,Fire,Fart and Tearing. From analysing the FSDnoisy18k [4], it can be inferred that the noise modelling layer improved accuracy in certain classes with significant amount of noisy labels, such as clapping (68% noisy labels), coin dropping (71% noisy labels), crash cymbal (86% noisy labels) and wind (75% noisy labels), however this is not the case with all the classes with high label noise. For some labels such as piano (60% noisy labels), Engine (68% noisy labels), Fire (89% noisy labels), the accuracy score of the noise modelling layer either dropped or stayed constant as compared to the baseline model.

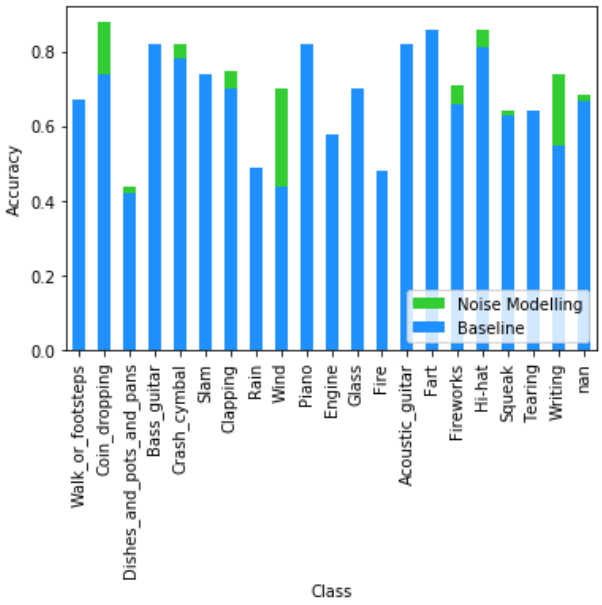

Figure 2: Classwise accuracy score for baseline model(blue) and noise modelling layer (green) is shown. Both graphs have been merged for the purpose of comparison. For all the classes with only blue bars, the accuracy achieved by the noise modelling layer is either equivalent or less than the baseline model and the green bar is a visual indication of the classes where the noise modelling layer was able to improve the accuracy of the baseline model.

This inconsistency in performance improvement indicates towards the hypothesis that the noise modelling approach might only be effective against certain label noise types and not so much against other types. We intend to explore this hypothesis in a more detailed manner in the future.

9. FUTURE WORK & CONCLUSION

In this work, we experimented with an intuitive approach to model the noise distribution of dataset labels and compared it with a noise robust loss function approach. The accuracy increase over the baseline model is encouraging, however we believe that a higher accuracy can be obtained by further tuning of the network and implementing the complex model from [6]. Although the accuracy of the system is lower than the one reported in [4], we consider this our first step in exploration of modelling label noise distribution and hope to achieve better results in the future.

From a future work perspective, we intend to understand as to why the noise modelling layer only can rectify certain kinds of label noise and fails to do so on other kinds of noise, post which we intend to implement the complex model from [6] to evaluate its performance against the baseline model, baseline with simple noise modelling layer and the soft bootstrapping loss model. Our future road map also includes implementing the approach on a multi-label noisy audio dataset and evaluate the performance of the model from different evaluation metrics other than accuracy to gain a better understanding of the underlying concepts.

10. REFERENCES

[1] M. D. P. T. Virtanen and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Springer, 2018.

[2] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, "CNN architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[3] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 2018.

[4] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, "Learning sound event classifiers from web audio with noisy labels," in *Proceedings of ICASSP 2019*, 2019.

[5] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[6] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *ICLR*, 2017.

[7] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[8] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 839–847.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society, series B*, 1977.

[10] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017.

[11] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, 2017.

[12] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," in *CVPR*, 2018.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] "DCASE 2019 Task2." [Online]. Available: http://dcase.community/challenge2019/task-audio-tagging

[16] "Imbalanced learning." [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/api.html

[17] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.